Ściąga do vi

Sposoby uruchomienia vi

W najprostszy sposób uruchomienie `vi`, polega na wpisaniu nazw plików poprzedzonych nazwą `vi`:

`vi` *plik plik*`...`

Użyteczna jest także opcja `+` określająca, w którym wierszu zostanie umieszczony kursor po uruchomieniu `vi`:

`vi +`*n plik* lub `vi +/`*tekst plik*

Pierwszy z w/w sposobów oznacza umieszczenie kursora w wierszu *n* (`$` oznacza ostatni wiersz); drugi sposób spowoduje ustawienie kursora w miejscu pierwszego wystąpienia *tekstu*. Wywołanie `view` zamiast `vi` powoduje uruchomienie `vi` w trybie tylko do odczytu (równoważne opcji `vi -R`).

Uruchomienie `vi` z opcją `-r` pozwala na odzyskanie zawartości pliku wymiany. Uruchomienie `vi -r` (bez nazwy pliku) spowoduje wyświetlenie listy odnalezionych plików wymiany. Zamiast uruchomienia `vi -r` można zastosować polecenie `:recover` *plik* (wersja skrócona `:rec` *plik*). Pominięcie nazwy *pliku* powoduje, że `vi` wczyta plik wymiany bieżącego pliku.

Podstawowe operacje edycyjne

Operacje na plikach są w większości realizowane przez polecenia rozpoczynające się od znaku dwukropka. Naciśnięcie tego znaku powoduje przejście do trybu poleceń edytora `ex`.

Podstawowe Operacje na plikach

`ZZ` lub `:x`	wyjście z zapisaniem do bieżącego pliku
`:w`	zapisanie zmian do bieżącego pliku
`:q`	wyjście, jeśli plik nie był modyfikowany
`:q!`	wyjście bez zapisywania zmian
`:w` *plik*	zapisanie zawartości bufora do *pliku*
`:`*x*`,`*y*`w` *plik*	zapisanie od wiersza *x* do *y* do *pliku*
`:e` *plik*	rozpoczęcie edycji *pliku*
`:e!`	pominięcie dokonanych modyfikacji i udostępnienie ponownie bieżącego *pliku* do edycji
`:r` *plik*	wstawienie zawartości *pliku* za bieżącym wierszem
`:n`	edycja następnego pliku z listy argumentów
`:N`	edycja poprzedniego pliku z listy argumentów

Polecenie `:n` (skrót od `:nextfile`) pozwala przejść do edycji następnego pliku tylko gdy bieżący został zapisany.

Polecenie `f:` *plik* nadaje bieżącemu plikowi nazwę *plik*. Równoważne polecenia `:ls`, `:files`, `:buffers` wyświetlają na ekranie listę edytowanych w danej chwili plików. Pierwsza kolumna na liście to numer bufora, który może być wykorzystany jako argument polecenia `:buffer` (w skrócie `:b`) lub *numer*`<Ctrl>-^`. Podanie `:b` *numer* (*numer*`<Ctrl>-^`) powoduje przejście do bufora *numer*.

Przejście do trybu wprowadzania tekstu

`i`	wstawianie tekstu przed kursorem
`I`	przesuwa kursor do początku wiersza bieżącego i przechodzi do trybu wpisywania
`o`	tworzy wiersz poniżej bieżącego, zaczyna tryb wpisywania
`O`	tworzy wiersz *powyżej* bieżącego, zaczyna tryb wpisywania
`a`	wstawianie tekstu za bieżącą pozycją kursora
`A`	przesuwa kursor na koniec wiersza bieżącego, zaczyna trybu wpisywania
`r`*c*	wstawienie znaku *c* w pozycji kursora
`R`	przejście do trybu nadpisywania tekstu
`<Esc>`	przejście do trybu wprowadzania poleceń

Przesuwanie kursora

`0` lub `^`	najbliższy początek wiersza z lewej*
`$`	najbliższy koniec wiersza z prawej*
`+` lub `<Enter>`	początek następnego wiersza
`-`	początek poprzedniego wiersza
`h`, `j`, `k`, `l`	odpowiednio: znak po lewej, poniżej, powyżej, po prawej*
`G`	początek ostatniego wiersza w pliku
`H`	kursor do górnego lewego rogu ekranu
`M`	kursor do środka ekranu
`L`	kursor do lewego dolnego rogu ekranu
`w`	początek najbliższego słowa z prawej*
`b`	początek najbliższego słowa z lewej*
`e`	koniec najbliższego słowa z prawej*
`(`	najbliższy początek zdania z lewej*
`)`	najbliższy początek zdania z prawej*
`{`	najbliższy początek akapitu z lewej*
`{`	najbliższy koniec akapitu z prawej*
`/`*napis*	wystąpienie *napisu* z prawej*
`?`*napis*	wystąpienie *napisu* z lewej*

* względem bieżącej pozycji kursora

Każde polecenie z powyższego zestawienia może być poprzedzone liczbą określającą zwielokrotnienie. Przykładowo: `1G` przesuwa kursor do pierwszego wiersza pliku, zaś *n*`G` do wiersza o numerze *n*, `2w` oznacza początek drugiego z kolei słowa z prawej strony względem pozycji kursora, `3{` określa początek trzeciego kolejnego akapitu. (Akapit to fragment tekstu oddzielony co najmniej jednym pustym wierszem). W poleceniach `w`, `b`, `e` słowo oznacza ciąg znaków alfanumerycznych bez znaków przestankowych. Polecenia `W`, `B`, `E` działają tak jak `w`, `b`, `e`, z tym, że słowo oznacza ciąg znaków otoczony znakami odstępu.

Przesuwanie kursora

`<CTRL>-f`	przesuwa tekst o 1 ekran do przodu (*forward*)
`<CTRL>-b`	przesuwa tekst o 1 ekran do tyłu (*backward*)
`<CTRL>-u`	przesuwa tekst o pół ekran do przodu
`<CTRL>-d`	przesuwa tekst o pół ekran do tyłu

Polecenia `<CTRL>-u` i `<CTRL>-d` oznaczają przesunięcie o pół ekranu w odpowiednim kierunku. Wielkość tę można zmieniać podając odpowiednią liczbę wierszy jako argument polecenia. Brak specyfikacji oznacza przyjęcie ostatnio podanej liczby wierszy.

Usuwanie i modyfikowanie tekstu

Do usuwania fragmentów tekstu służy polecenie `d`*zakres*, gdzie *zakres* określa, co zostanie usunięte. *Zakres* określamy dodając do polecenia `d` polecenie przesunięcia kursora. „Nietypowa” wersja `dd` polecenia `d` usuwa cały wiersz.

Polecenie `c` usuwa fragment tekstu określony przez *zakres*, a następnie przechodzi do trybu wstawiania. Polecenie `c` jest równoważne sekwencji `d`*zakres* `i` `a`.

Usuwanie i modyfikowanie tekstu

n`x`	usuwa *n* znaków na prawo od kursora
n`X`	usuwa *n* znaków na lewo od kursora
`D` lub `d$`	usuwa od kursora do końca wiersza
`dd`	usuwa bieżący wiersz
n`dd` lub `d`*n*`d`	usuwa *n* wierszy począwszy od bieżącego
`d0`	usuwa od początku wiersza do kursora
`dw`	usuwa od kursora do początku następnego wyrazu
`dH`, `dM`, `dL`	usuwa od kursora odpowiednio do góry, środka i dołu ekranu
`dG`	usuwa od bieżącego wiersza do końca pliku
`d/`*napis*`<Enter>`	usuwa od kursora do podanego *napisu* (*napis* jest wyrażeniem regularnym)
`cc`*tekst*`<Esc>`	zamienia bieżący wiersz na *tekst*
`C`*tekst*`<Esc>`	zamienia od kursora do końca wiersza na *tekst*

Sekwencja `ddp` zamienia miejscami wiersz bieżący z następnym.

Usuwany fragment tekstu jest kopiowany do specjalnego bufora (bufor nie nazwany). Zawartość tego bufora może zostać wstawiona w innym miejscu za pomocą polecenia `p` lub `P`. Polecenie `y`, o składni podobnej do poleceń `d` czy `c`, kopiuje fragment tekstu określony przez *zakres* do bufora nie nazwanego.

Przesuwanie tekstu z/do bufora tymczasowego

`d`, `dd` lub `D`	usuwa tekst do bufora nie nazwanego
`yy` lub `Y`	kopiuje bieżący wiersz do bufora nie nazwanego
n`yy` lub *n*`Y`	kopiuje *n* wierszy do bufora nie nazwanego, począwszy od bieżącego
`y`*n*`w`	kopiuje *n* kolejnych słów do bufora nie nazwanego
`y/`*tekst*	kopiuje od kursora do *tekstu* do bufora nie nazwanego
`p`	wstawia zawartość bufora nie nazwanego za wiersz bieżący
`P`	jak wyżej, ale przed wiersz bieżący

Operacje wyszukiwania i zamiany

Poleceniem wyszukiwania jest `/`. Po naciśnięciu `/`, znak ten pojawia się w ostatnim wierszu ekranu, a `vi` czeka na podanie szukanego tekstu. Po wpisaniu tekstu należy nacisnąć `<Enter>`. Tekst jest szukany od bieżącej pozycji kursora do końca pliku. Po napotkaniu końca pliku poszukiwanie jest kontynuowane od jego początku.

Znak `/` oznacza wyszukiwanie w kierunku końca pliku, natomiast `?` w kierunku początku pliku. Ostatnie wyszukiwanie można powtórzyć za pomocą polecenia `n` (albo naciskając `/` (lub `?`) i `<Enter>`).

Zamiana jest realizowana za pomocą polecenia `:s`. W najprostszej postaci polecenie to postać:

`:s/`*tekst*`/`*tekst-do-wymiany*`<Enter>`

Instrukcja taka działa tylko w obrębie bieżącego wiersza i zamienia pierwsze wystąpienie *tekstu* na *tekst-do-wymiany*. Polecenie zamiany można poprzedzić określeniem zakresu wierszy, których ma dotyczyć. Zakres to dwa numery wierszy oddzielone przecinkiem, np. `10,12`. Podane numery są bezwzględnymi numerami wierszy w pliku (polecenie `<Ctrl>-g` wyświetla numer bieżącego wiersza). Ostatni wiersz w pliku jest oznaczamy symbolem `$`, zaś bieżący symbolem `.` (kropka). Cały plik oznaczamy jako `1,$` lub używając znaku `%`. Na końcu polecenia `s` umieszczamy opcje. Podanie opcji `g` oznacza, że zmiany mają dotyczyć wszystkich wystąpień zamienianego *tekstu* w wierszu. Opcja `/c` powoduje wykonywanie zamiany z potwierdzeniem.

Znak-separator `/` może być zastąpiony innym. Ogólnie, pierwszy znak za nazwą polecenia `s` jest interpretowany jako znak separujący. Znak separujący wykorzystany dosłownie w *tekście* musi być poprzedzony znakiem `\`.

Operacje wyszukiwania i zamiany

`/`*napis*`<Enter>`	szuka (do przodu) ciągu znaków *napis* od kursora do pierwszego znalezionego lub do końca pliku, a następnie od początku pliku do wiersza bieżącego
`?`*napis*`<Enter>`	szuka jak wyżej, ale do tyłu
`n` lub `N`	powtórzenie ostatniego wyszukiwania w tym samym kierunku (`n`), lub w przeciwnym (`N`)
`:`*m*`,`*n*`s/`*x*`/`*y*`/`*o*	w wierszach od *m* do *n* zamienia napis *x* na *y* (*o* oznacza opcje)

Przy wyszukiwaniu/zamianie możliwe jest wykorzystanie wyrażeń regularnych. Zmiana z potwierdzeniem: `y<Enter>` zamienia, `<Enter>` pomija.

Wyrażenia regularne edytora `vi`

Wyrażenie	Znaczenie
c	znak nie będący metaznakiem
c	znak/metaznak *c*
`.`	dowolny znak (z wyjątkiem końca wiersza)
`^`	początek wiersza
`$`	koniec wiersza
`\<`	początek słowa
`\>`	koniec słowa
`[ab...]`	dowolny ze znaków `a`, `b`...
`[^ab...]`	dowolny ze znaków oprócz `a`, `b`...
`[a-z]`	dowolny ze znaków z zakresu `a-z`
`[^a-z]`	dowolny ze znaków oprócz `a-z`
`\(`*r*`\)\{m,n\}`	od *m* do *n* napisów pasujących do *r* (*r* oznacza wyrażenie regularne)
`\(`*r*`\)\{m,\}`	co najmniej *m* powtórzeń *r*
`\(`*r*`\)\{,n\}`	zero lub co najwyżej *n* powtórzeń *r*
`\(`*r*`\)\{n\}`	*n* powtórzeń *r*
r`*`	zero lub więcej napisów pasujących do *r*
`\(`*r*`\)`	operator grupowania
`\n`	*n*-ta zdefiniowana grupa (*n* = 1,...,9)
`&`	cały zastępowany napis (polecenie `s`)
`\u`	zamienia następny znak na dużą literę
`\l`	zamienia następny znak na małą literę
`\U`	zmienia kolejne znaki na duże litery do końca napisu lub napotkania `\E` (`\e`)
`\L`	zmienia kolejne znaki na małe litery do końca napisu lub napotkania `\E` (`\e`)

`&`, `\u`, `\l`, `\U`, `\L` mogą się pojawić tylko w *tekście-do-wymiany*, oto przykłady:

```
:%s/^.*$/&&&
:%s/\<[a-e]..\>/\U&/g
:%s/\<([A-E])(..)\>/\l\1\U\2/g
```

Pierwsze polecenie spowoduje trzykrotne powielenie zawartości każdego wiersza. Drugie zastępuje każde trzyznakowe słowo zaczynające się od liter a–e, na to samo słowo ale pisane dużymi literami. W trzecim przykładzie, w każdym trzyznakowym słowie zaczynającym się od dużej litery A–E, pierwsza litera zostanie zmieniona na małą a pozostałe będą zamienione z małych na duże.

Polecenie `f`*c* wyszukuje znak *c* od kursora do końca *bieżącego wiersza*, `F`*c* wyszukuje *c* od kursora do początku bieżącego wiersza. Polecenie `;` (średnik) powtarza ostatnie wyszukiwanie, natomiast `,` (przecinek) powtarza ostatnie wyszukiwanie w odwrotnym kierunku.

Działania na buforach nazwanych

Edytor `vi` pamięta więcej niż tylko ostatnio usunięty fragment. Pamiętanych jest dziewięć ostatnio przeniesionych poleceniami `d`, `c` lub `y` bloków, które są umieszczane automatycznie w buforach, ponumerowanych od `1` do `9`. Bufor `1` (tj. bufor nie nazwany) zawiera ostatnio usunięty/skopiowany blok. Kolejne wykonanie poleceń `d`, `c`, `y` powoduje przeniesienie fragmentów tekstu do buforów o wyższych numerach. Aby odzyskać tekst z określonego bufora, należy wykonać polecenie `p` lub `P`, poprzedzając je znakiem `"` oraz numerem, np. `"3p` wstawia zawartość bufora o numerze 3.

Ponadto istnieje 26 buforów o jednoliterowych nazwach, od `a` do `z`. Aby wstawić tekst do bufora nazwanego, odpowiednie polecenie należy poprzedzić znakiem `"` oraz nazwą bufora, np. `"add` umieszcza usuwany wiersz w buforze `"a`. Przeglądanie zawartości buforów realizuje następująca sekwencja poleceń: `"1p` a następnie sukcesywnie `u.`.

Nazwy buforów muszą być zapisane małymi literami. Użycie dużej litery powoduje, że tekst zostanie *dołączony* do zawartości bufora. W przypadku posługiwania się małymi literami tekst przesyłany zastępuje dotychczasową zawartość bufora.

Działania na buforach nazwanych

`"`*x*`ndd`	usuwa z tekstu *n* wierszy i przesyła do bufora *x*
`"`*x*`yy`	kopiuje bieżący wiersz do bufora o nazwie *x*
`"`*x*`p`	wstawia zawartość bufora *x* za bieżący wiersz
`"`*x*`P`	wstawia zawartość bufora *x* przed bieżący wiersz

Polecenie `m` (*mark*) służy do oznaczania, za pomocą jednoliterowych znaczników dowolnych miejsc w dokumencie. Poniżej zestawiono przykłady użytecznych poleceń wykorzystujących znaczniki:

Bloki tekstu oznakowane znacznikami

`m`*a*	oznakowuje znacznikiem *a* (*a* jest dowolną małą literą) miejsce wskazane kursorem
`` `a ``	powrót kursora do znacznika *a*
`'a`	powrót kursora do wiersza oznaczonego znacznikiem *a*
`d'`*a*	usuniecie bloku tekstu od wiersza ze znacznikiem *a* do bieżącej pozycji kursora i przesłanie go do bufora nie nazwanego
`"`*z*`d'`*a*	jak wyżej ale blok jest usuwany do bufora *z*
`` `` ``	powrót do poprzedniego położenia kursora
`''`	powrót do wiersza, w którym znajdował się uprzednio kursor
`:'`*a*`,'`*b*`m.`	przesuwa oznaczone wiersze, od *a* do *b*, za wiersz bieżący
`:'`*a*`,'`*b*`d`	usuwa oznaczone wiersze od *a* do *b*
`:'`*a*`,'`*b*`w` *plik*	zapisuje oznaczone wiersze, od *a* do *b*, do nowego *pliku*
`:'`*a*`,.w` *plik*	zapisuje wiersze, od oznaczonego znacznikiem *a* do bieżącego, do nowego *pliku*
`:'`*a*`,'`*b*`w!` *plik*	nadpisuje istniejący *plik* oznaczonymi wierszami od *a* do *b*
`:'`*a*`,'`*b*`w<<`*plik*	oznaczone wiersze, od *a* do *b*, dopisuje na koniec *pliku*

Pozostałe użyteczne polecenia

Interakcja z powłoką

`:!`*polecenie*	uruchamia *polecenie* systemowe
`:`*n1*`,`*n2*`!`*polecenie*	blok od wiersza *n1* do *n2*, jest filtrowany przez *polecenie*
`:'a,'b!`*polecenie*	oznaczone wiersze, od `a` do `b`, są filtrowane przez *polecenie*
`:sh`	wywołuje nową powłokę; powrót po EOF

Inne polecenia

`J`	łączy wiersz bieżący z następnym
n`<<`	przesuwa *n* kolejnych wierszy w lewo o 1 znak
n`>>`	jak wyżej, ale w prawo
`u`	cofa ostatnie polecenia (*undo*)
`%`	szuka znaku do pary `{}`, `()` lub `[]`
`U`	odtwarza stan bieżącego wiersza (*undo*)
`<Ctrl>-g`	podaje numer bieżącego wiersza w pliku
`.`	(kropka) powtarza ostatnie polecenie

`:so` *plik*	wykonuje polecenie edytora z *pliku*

Podział na okna (tylko `vim`): `<Ctrl>ws` (albo `:split`). Zamknięcie okna, w którym jest kursor: `:q`. Przejście do okna poniżej bieżącego: `<Ctrl>wj` (`<Ctrl>wk`, powyżej okna bieżącego).

Skróty i makrodefinicje

Skróty działają w trybie wprowadzania tekstu. Każdy skrót umieszczony w tekście zostanie automatycznie rozwinięty do pełnej formy. Oto przykładowa definicja:

```
:ab PDF Portable Document Format
```

Rozwinięcie, nastąpi tylko wówczas, gdy skrót pojawi się jako oddzielne słowo, a nie jako fragment innego wyrazu. Definicję skrótu można anulować poleceniem, np. `:una PDF`.

Makrodefinicje odgrywają podobną rolę jak skróty, z tym, że działają w trybie poleceń. Oto przykład:

```
:map !! 1Gi#! /usr/bin/awk -f<Esc>o
```

Każde naciśnięcie dwóch `!!` spowoduje przejście do pierwszego wiersza w pliku (`1G`), przejście do trybu wpisywania (`i`), wstawienie napisu `#! /usr/bin/awk -f`, przejście do trybu poleceń (`<Esc>`), rozpoczęcie trybu wpisywania w nowym wierszu. Makrodefinicje można anulować za pomocą polecenia `:unmap`.

Konfigurowanie edytora

Działanie edytora można konfigurować nadając odpowiednie wartości zmiennym globalnym za pomocą polecenia `:set`. Część zmiennych globalnych to przełączniki. Przykładowo, aby włączyć przełącznik `ic` (rozróżnianie małych/dużych liter), wydajemy polecenie `:set ic`. Wyłączamy poleceniem `set: noic` (każdy przełącznik wyłączamy przez dodanie przedrostka `no` do nazwy zmiennej).

Wybrane zmienne globalne

Zmienna	Znaczenie
`ai`*	automatyczne wcinanie wiersza na głębokość wcięcia wiersza poprzedniego
`dir`	określenie katalogu dla plików wymiany
`ic`*	nie rozróżnianie dużych/małych liter
`list`*	oznaczanie na ekranie końca wiersza znakiem `$`
`nu`*	numerowanie wierszy
`ts`	rozmiar wcięcia wykonywanego klawiszem `<Tab>`
`wm`	jeżeli różne od zera, to `vi` automatycznie justuje wiersze
`showmode`*	wyświetlanie trybu pracy
`magic`*	traktowanie `.[*` jako metaznaków w wyszukiwaniu i zamianie

* przełącznik

Jeżeli chcemy używać edytora `vi` z własnym zestawem zmiennych globalnych, to polecenia `set` (bez dwukropka na początku) należy umieścić w pliku `.exrc`. Plik ten powinien się znajdować w katalogu domowym lub w katalogu bieżącym (zestaw przeszukiwanych katalogów zależy od konfiguracji `vi`).

Opracowanie: Tomasz Przechlewski